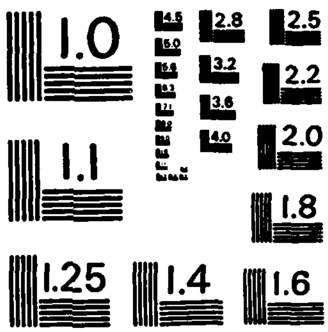
AD-A121 507 AN INTERACTIVE PROGRAM FOR INVENTORY TREND ANALYSIS USING NONPARAMETRIC S..(U) NAVAL AIR DEVELOPMENT CENTER WARMINSTER PA SYSTEMS DIRECTORAT.. D BIRNBAUM ET AL. 1/1

UNCLASSIFIED 01 OCT 82 NADC-82246-20 MIPR-2-0007 F/G 9/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

REPORT NO. NADC-82246-20

AN INTERACTIVE PROGRAM FOR INVENTORY TREND ANALYSIS USING NONPARAMETRIC STATISTICAL TECHNIQUES

by

D. Birnbaum
and
R. Doray
Systems Directorate
NAVAL AIR DEVELOPMENT CENTER
Warminster, Pennsylvania 18974

AD A121507

1 OCTOBER 1982

FINAL REPORT
TASK NO. MIPR 2-0007

Approved for Public Release; Distribution Unlimited

Prepared for
NAVAL AIR SYSTEMS COMMAND
Department of the Navy
Washington, D. C. 20361

NOTICES

REPORT NUMBERING SYSTEM – The numbering of technical project reports issued by the Naval Air Development Center is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Command Office or the Functional Directorate responsible for the report. For example: Report No. NADC-78015-20 indicates the fifteenth Center report for the year 1978, and prepared by the Systems Directorate. The numerical codes are as follows:

CODE	OFFICE OR DIRECTORATE
00	Commander, Naval Air Development Center
01	Technical Director, Naval Air Development Center
02	Comptroller
10	Directorate Command Projects
20	Systems Directorate
30	Sensors & Avionics Technology Directorate
40	Communication & Navigation Technology Directorate
50	Software Computer Directorate
60	Aircraft & Crew Systems Technology Directorate
70	Planning Assessment Resources
80	Engineering Support Group

PRODUCT ENDORSEMENT – The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

APPROVED BY: R.A. Stampfl DATE: 10/8/82

R. A. STAMPFL
DIRECTOR, SYSTEMS DIRECTORATE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER NADC-82246-20	2. GOVT ACCESSION NO. AD-A121 507	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN INTERACTIVE PROGRAM FOR INVENTORY TREND ANALYSIS USING NONPARAMETRIC STATISTICAL TECHNIQUES		5. TYPE OF REPORT & PERIOD COVERED FINAL
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) DAVID BIRNBAUM RONALD L. DORAY		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS NAVAL AIR DEVELOPMENT CENTER SYSTEM DESIGN - 203 WARMINSTER, PENNSYLVANIA 18974		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS MIPR 2-0007
11. CONTROLLING OFFICE NAME AND ADDRESS NAVAL AIR DEVELOPMENT CENTER WARMINSTER, PENNSYLVANIA 18974		12. REPORT DATE 1 OCT 1982
		13. NUMBER OF PAGES 24
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) DEFENSE INDUSTRIAL SUPPLY CENTER 700 ROBBINS AVENUE PHILADELPHIA, PENNSYLVANIA 19111		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

TREND ANALYSIS
NONPARAMETRIC STATISTICS
KENDALL RANK CORRELATION STATISTIC

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes an interactive program developed for inventory control. The program determines if there is a statistically significant trend in inventory over a number of years by utilizing nonparametric statistics to test if the ranked inventory data are random against a monotonic trend.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

Introduction

Inventory maintenance is very critical in maintaining fleet readiness. There is a cost in maintaining a set inventory and a cost in not being able to supply spare parts for existing assets. Therefore, the Naval Supply Depot, under Task Order MIPR2-0007, requested NAVAIRDEVCEN to develop an interactive computer program that determines if there is a statistically significant positive or negative trend in inventory over a number of years. A negative trend would of course require a reordering of additional spare parts while a positive trend would indicate a reduction in spare parts, thereby reducing the costs of inventory maintenance.

The program developed and discussed in this report utilizes nonparametric statistics to test if the inventory changes are random against a monotonic trend. When the set of yearly inventories are ranked and tested against the inventory years, the agreement or disagreement with a positive yearly trend determines if the inventory has increased or decreased over the years, i.e., a large positive (negative) number would indicate a positive (negative) trend while a number close to zero would indicate no inventory change.

Mathematical Background

The classical measure of association between two variables is the correlation coefficient. The importance of the correlation coefficient arises from parametric linear regression theory. However, in non-normal populations, dependence among variables can manifest itself in other ways. For example, the Kendall Rank Correlation coefficient can be used to determine the degree of relationship between populations and is dependent on the value of parameter "S". The computation of S is performed by counting how often the two rankings from different variables move in the same direction and how often they move in the opposite direction. This computation is performed for all possible pairs in the number sequences. The strength of the relationship is obtained by considering S relative to its two extremes, forming the quantity "t" known as the Kendall Rank Correlation Coefficient. A value of t near +1 implies close agreement between the rankings while a value of t near -1 implies opposite rankings. A value of t in the neighborhood of zero indicates neither agreement nor disagreement (reference (a)).

Since many of the values derived for t and consequently S do not differ significantly from zero, a statistical test must be applied to determine when t differs significantly from zero. This test is known as the test of "independence" and is defined in reference (a).

A statistic z can be developed to test the hypothesis H_o that the two variables: years and the ranking for the inventory, are independently distributed, i.e., the test statistic S does not significantly differ from zero implying that neither a positive nor negative trend exists. The test statistic z for a sample with no ties and sample size of at least eight samples can be approximated by a normal distribution and is given by:

$$z = \frac{6\,S}{2n(n-1)(2n+5)}$$

The program as defined in the subroutine TAU1 will compensate the test statistic z for ties and requires a minimum of six samples.

Program Description

The trend analysis program developed is an interactive program written to guide the user through the program. Following the sign-on procedure given in Appendix A the user loads the program from the library with the command:

)LOAD TZIMES

The trend analysis program can be activated by the command TREND and "return". The program will then request the user to enter the FSC # shown below

```
ENTER FSC NUMBER
□:
34
```

The #34 was entered and the program responded with

```
FSC NUMBER =34
```

The computer then requested the inventory to be entered for each year by

```
ENTER INVENTORY DEMAND FROM FIRST YEAR TO PRESENT
□:
```

The inventory 495 through 550 was then entered into the program with the depression of the return key

```
495 525 520 490 555 530 475 510 515 545 540 550
```

The program then requested the user to enter the corresponding years by printing

```
ENTER INVENTORY YEARS
□:
```

The years entered must be in increasing order and correspond to the inventory entered. For this problem the following years were entered

```
1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981
```

The program then analyzes the entered data and prints; the Kendall rank correlation coefficient, the S value and the descriptive level of the test, where this level describes the area under the normal curve to the right of the computed z value defined above. The smaller this value is, the lower the probability of saying there is a positive trend or that there is no trend (Type II Error). The probability of a type two error is 8.5%.

```
CORRELATION COEFFICIENT=0.303030303
TREND IS POSITIVE S=20
DESCRIPTIVE LEVEL OF TEST =0.08511707990
```

The program then requests the user to respond to the following cue

```
ENTER 2 IF YOU WISH TO OBTAIN LINE OF REGRESSION OTHERWISE ENTER 1
?:
      2
```

A response of 1, skips this part of the program while a response of 2 results in the following:

1. The program fits a straight line to the inventory as a function of the entered years. This program is listed and described in Appendix B.

2. The coefficients A and B of the linear equation Y = X + BX are computed and listed.

3. The standard error of the regressed line is computed and listed.

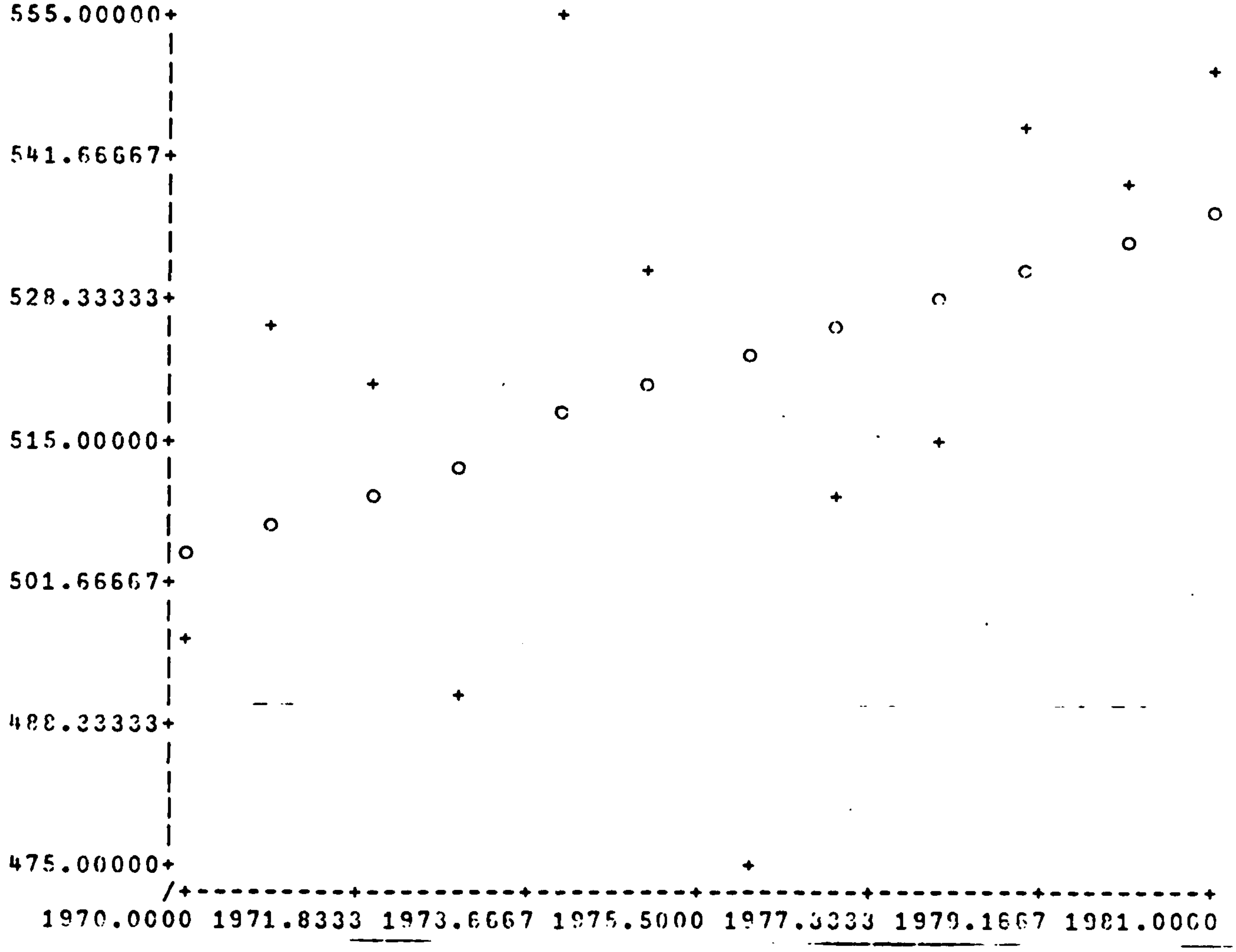
4. The correlation coefficient is computed and listed.

5. The T value from which the confidence can be computed. These outputs are shown below.

```
THE STRAIGHT LINE IS Y=A+BX WITH A AND B =¯5350.407925 2.972027972
STANDARD ERROR OF ESTIMATE =0.4265687474
SIMPLE CORRELATION COEFFICIENT =0.4265687474
T VALUE =1.491427002
```

The program then plots the regressed line (circles) and the inventory (plusses). The scatter of the inventory data relative to the regressed line indicates graphically the goodness of fit.

```
VALUE =1.491427002
```



If the user wishes to project this linear trend into the future, the cue response to "Inventory Prediction" should be "2" and return.

```
ENTER 2 IF INVENTORY PREDICTION IS DESIRED OTHERWISE ENTER 1
⎕:
      2
```

The program then requests the years prediction to be entered. In response to the entry the program computes and lists the inventory prediction for the entered years.

```
ENTER YEARS OF PREDICTION
⎕:
      1982 1983 1984
INVENTORY =540.1515152 543.1235431 546.0955711
```

The program can then be cycled or ended by the response of 2 or 1 to the next inquiry.

```
ENTER 2 IF MORE DATA IS TO BE ANALYZED OTHERWISE ENTER 1
[illegible]:
        2
```

Summary/Conclusion

The program described in this report calculates the Kendall Rank Correlation coefficient which is combined with a test for independence against a monotonically increasing function. This problem will be sensitive to changing trends when several years are tested, but will tend to be insensitive to short term inventory changes.

APPENDIX A

Main Program - TREND

The function "TREND" is the main program for the Trend Analysis program and is listed below.

```
      ∇TREND [⎕]∇
      ∇TREND
[1]    SP←6⍴0
[2]   RETURN1:'ENTER FSC NUMBER'
[3]    'FSC NUMBER =';⎕
[4]    'ENTER INVENTORY DEMAND FROM FIRST YEAR TO PRESENT'
[5]    SET←⎕
[6]    Y←SET
[7]    'ENTER INVENTORY YEARS'
[8]    YR←⎕
[9]    →((⍴YR)≠(⍴Y))/ERROR
[10]   ORDER Y
[11]   TAU1
[12]   'CORRELATION COEFFICIENT=';(2×S)÷(N×(N-1))
[13]   →(S<0)/PROB
[14]   'TREND IS POSITIVE S=';S
[15]   'DESCRIPTIVE LEVEL OF TEST =';1-(NCUM Z)
[16]  RETURN2:'ENTER 2 IF YOU WISH TO OBTAIN LINE OF REGRESSION OTHERWISE ENTE
[17]   URN←⎕
[18]   →(URN=1)/RETURN3
[19]   T←YR SR SET
[20]   LINREG←(T[4;1]×YR)+T[3;1]
[21]   'THE STRAIGHT LINE IS Y=A+BX WITH A AND B =';T[3;1],T[4;1]
[22]   'STANDARD ERROR OF ESTIMATE =';T[5;1]
[23]   'SIMPLE CORRELATION COEFFICIENT =';T[5;2]
[24]   'T VALUE =';T[4;3]
[25]   MAXLIN←⌈/(⌈/LINREG,⌈/SET)
[26]   MINLIN←⌊/(⌊/LINREG,⌊/SET)
[27]   SP[1]←30
[28]   SP[2]←MINLIN
[29]   SP[3]←(MAXLIN-MINLIN)÷6
[30]   SP[4]←60
[31]   SP[5]←⌊/YR
[32]   SP[6]←(⌈/YR-⌊/YR)÷6
[33]   SP PLOT LINREG AND SET VS YR
[34]   'ENTER 2 IF INVENTORY PREDICTION IS DESIRED OTHERWISE ENTER 1'
[35]   URN2←⎕
[36]   →(URN2=1)/RETURN3
[37]   'ENTER YEARS OF PREDICTION'
[38]   PREY←⎕
[39]   'INVENTORY =';(T[4;1]×PREY)+T[3;1]
[40]  RETURN3:'ENTER 2 IF MORE DATA IS TO BE ANALYZED OTHERWISE ENTER 1'
[41]   URN1←⎕
[42]   →(URN1=1)/0
[43]   →RETURN1
[44]  ERROR:'INCONSISTENCY BETWEEN DATA BASE AND YEARS'
[45]   →RETURN1
[46]  PROB:'TREND IS NEGATIVE S=';S
[47]   'DESCRIPTIVE LEVEL OF TEST =';1-(NCUM|Z)
[48]   →RETURN2
      ∇
```

STATEMENT #	FUNCTION
[1]	Shapes a row vector SP
[2]	Requests user to enter FSC #
[3]	Prints entered FSC #
[4]	Requests inventory
[5]	Assigns the variable SET equal to the inventory
[6]	Assigns the variable Y equal to SET
[7]	Requests inventory years
[8]	Assigns the variable YR to inventory years
[9]	Checks that each year has an inventory - if there is an inconsistency, a skip to statement [44] occurs
[10]	The function ORDER is called and the rank for the inventory is determined as a function of the years
[11]	The function TAU1 is then called and the variable S is determined
[12]	Kendall Rank correlation coefficient is computed
[13]	"SIGN" of S is determined and if negative skips to statement [46]
[14]	Positive value of S is noted and value is printed
[15]	Descriptive level of test - (area under normal curve to the right of "Z")
[16]	Open the option of user to obtain coefficients for linear equation and obtain for plot
[17]	URN - is the response variable of statement [16]
[18]	Test of statement with skip to statement [40] if one is entered

STATEMENT #	FUNCTION
19	Call for regression SR that computes coefficients and fits statistics
20	LINREG is the linear equation for the line with coefficients computed in 19
21	Coefficients A and B are printed for curve Y = A + BX
22	The standard error of the estimate is printed
23	The correlation coefficient is printed
24	The T value is printed
25	Maximum value of the regressed curve is determined
26	Minimum value is determined for regressed curve
27 - 32	Scaling is performed for the plot function
33	The plotting function is called and the data vs Y are plotted
34	Prediction is selected by user or program skips to 40
35	URN 2 is the variable name for entry into prediction
36	The variable URN 2 is tested for 1 or 2
37	Request for years to which prediction is desired
38	PREY - variable name for the prediction years
39	Inventory is computed and outputted
40	Enter new set of data or end program
41	Variable name URN 1 for response to item 40
42	Test of variable URN; of 1, program goes to logical zero which is end of program
43	Go to statement # 2

STATEMENT #	FUNCTION
[44]	Go to statement when there is inconsistency between YR and SET variable
[45]	Go to statement # [2]
[46]	Go to statement when S is negative
[47]	Descriptive level of test for negative Z
[48]	Go to statement # [2]

APPENDIX B

Subroutine - ORDER

This function ranks the data entered into the program and counts the number of ties. The variable to be ranked is X.

```
      ∇ORDER [□]∇
      ∇ORDER X;V
[1]    K←1
[2]    TIE←0
[3]    Y←0×(ιρX)
[4]   RETURN:V←(X=⌊/X)/X
[5]    →((ρV)≥2)/ADJUST
[6]    P←XιV
[7]    Y[P]←K
[8]    X[P]←999
[9]    K←K+1
[10]   →(K>ρX)/0
[11]   →RETURN
[12]  ADJUST:L←1
[13]   TIE←TIE+1
[14]  RETURN1:P←XιV[1]
[15]   K1←1÷(ρV)
[16]   Y[P]←K+K1
[17]   X[P]←999
[18]   L←L+1
[19]   →(L>ρV)/ADJUST1
[20]   →RETURN1
[21]  ADJUST1:K←K+(ρV)
[22]   →(K>ρX)/0
[23]   →RETURN
```

STATEMENT #	FUNCTION
[1]	Initialization of index K
[2]	Initialization of the variable TIE
[3]	Initialization of the row vector Y with length of the input variable X
[4]	V is a row vector containing the minimum elements of the vector X
[5]	If V contains two values i.e., there is a tie, the program jumps to statement [12]

STATEMENT #	FUNCTION
[6]	P is index variable for the element V in the vector X
[7]	The element P of vector Y is set equal to K
[8]	The element P of vector X is set equal to a large number
[9]	The index K is incremented
[10]	End program if the index K is greater than the # of elements in X
[11]	Go to statement [4]
[12] - [23]	The subroutine for ties within the matrix elements
[12]	Set variable L equal to one
[13]	TIE is incremented
[14]	Set the index P to the first element of V
[15]	K1 is the fraction to be added to K for the rank if more than one element are ranked the same
[16]	The rank of Y[P] is set
[17]	The variable X[P] is set to a large #
[18]	Increment the variable L
[19]	Go to statement [21] if all of the elements that are equal are set to the same rank
[20]	Go to statement [14]
[21]	Set the rank # K to the next value which equals the last rank plus ties
[22]	End program if all elements in X are complete
[23]	Go to statement #[4]

APPENDIX C

Subroutine - TAU1

The variable S, calculated by the subroutine, determines the number of times the ranking moves in the same or opposite direction from the monotomic function.

```
      ∇TAU1 [⎕]∇
      ∇TAU1
[1]    TU←2
[2]    L←0
[3]    S←0
[4]    N←⍴Y
[5]    K←0
[6]    ND1←0
[7]    NC1←0
[8]   RETURN:K←K+1
[9]    P←K+⍳(N-K)
[10]   R←⍴P
[11]   NC←+/((R⍴Y[K])<Y[P])
[12]   ND←+/((R⍴Y[K])>Y[P])
[13]   ND1←ND1+ND
[14]   NC1←NC1+NC
[15]   S←S+(NC-ND)
[16]   →(K=N)/CON
[17]   →RETURN
[18]  CON:VARS1←N×(N-1)×((2×N)+5)
[19]   →(TIE<1)/CONFIDENCE
[20]  ADJUST:L←L+(TU×(TU-1)×((2×TU)+5))
[21]   →(TU=TIE)/CONFIDENCE
[22]   TU←TU+1
[23]   →ADJUST
[24]  CONFIDENCE:VARS←(VARS1-L)÷18
[25]   SIGS←VARS*0.5
[26]   Z←S÷SIGS
[27]   →0
      ∇
```

STATEMENT #	FUNCTION
[1]	Initialize variable TU
[2]	Initialize variable L
[3]	Initialize variable S
[4]	Set the variable N equal to number of element in the row vector Y

STATEMENT #	FUNCTION
[5]	Initialize variable K
[6]	Initialize variable ND1
[7]	Initialize variable NC1
[8]	Increment variable K
[9]	Set P as the monotomic function with the same number of elements as X
[10]	R is the number of elements in P
[11]	Compute the number of elements that increase monotonically in Y i.e., are concordant
[12]	Compute the number of elements that decrease monotonically in Y i.e., are discordant
[13]	Counter for the number of discordant values
[14]	Counter for the number of concordant values
[15]	S is the total difference between concordant and discordant values
[16]	If the index K is equal to the number of elements in Y go to statement [18]
[17]	Go to statement [8]
[18]	Compute the variance of S
[19]	If no ties in the ranks go to statement [24]
[20]	Compute a correction for the variance of S for ties
[21]	If the variable TU equals the number of ties go to statement [24]
[22]	Increment the variable TU
[23]	Go to statement [20]
[24]	Compute the variance of S
[25]	Compute the standard deviation of S
[26]	Compute the statistic Z with N(0,1)
[27]	End program

APPENDIX D

Subroutine - NCUM 2

This subroutine is an approximation for the inverse normal distribution.

```
      ∇ERF [⎕]∇
      ∇PROB←ERF Z;TMP
[1]    TMP←(4.30638E¯5×Z*6)+(0.0002765672×Z*5)+(0.0001520143×Z*4)
+(0.0092705272×Z*3)+(0.0422820123×Z*2)+(0.07052307839998×Z)+1
[2]    PROB←1-1÷TMP*16
```

APPENDIX E

Subroutine - SR

Subroutine SR is part of the scientific subroutine package in the APL library. The following documentation is available for this program. A description: SRHOW and a program listing SR.

```
      SRHOW

    SIMPLE REGRESSION
    T←X SR Y
    ENETRED:04/16/70

   X AND Y ARE VECTORS GIVING THE (SAME NUMBER OF)
 OBSERVATIONS ON AN INDEPENDENT VARIABLE X AND A
 DEPENDENT VARIABLE Y. T IS A MATRIX WITH 5 ROWS
 AND 3 COLUMNS CONTAINING THE RESULT OF FITTING
 THE STRAIGHT LINE Y=A+B×X BY THE METHOD OF LEAST
 SQUARES IN THE FOLLOWING FORMAT:
    ROW1: MEAN OF X, ST DEV OF X, 0
    ROW2: MEAN OF Y, ST DEV OF Y, 0
    ROW3: A, 0, 0
    ROW4: B, ST ERROR OF B, T-VALUE
    ROW5: ST ERROR OF ESTIMATE, R=SIMPLE CORR COEFF, R*2
```

```
      ∇SR [⎕]∇
      ∇T←X SR Y
[1]    SX←((A←+/(X-MX←(+/X)÷N)*2)÷(N←(⍴X))-1)*0.5
[2]    SY←((B←+/(Y-MY←(+/Y)÷N)*2)÷N-1)*0.5
[3]    B0←MY-MX×B1←(+/(X-MX)×(Y-MY))÷A
[4]    SE←((B×1-RSQ←(R←B1×SX÷SY)*2)÷N-2)*0.5
[5]    TV←B1÷SB1←(SY÷SX)÷((N-2)÷(1-RSQ))*0.5
[6]    T←(5 3)⍴MX,SX,0,MY,SY,0,B0,0 0,B1,SB1,TV,SE,R,RSQ
      ∇
```

APPENDIX F

Subroutine - PLOT

The plotting routine is also part of the sceintific subroutine in the APL library. The Plothow function describes its use and capability and the program is listed.

```
PLOTHOW

GRAPHING FUNCTION:    G←S PLOT D

      DEPENDENT VARIABLES ARE PLOTTED AGAINST AN INDEPEN-
DENT VARIABLE.  IF THE RIGHT ARGUMENT D IS A VECTOR OR A
1-ROW MATRIX THE ELEMENTS OF D ARE CONSIDERED TO BE VALUES
OF A DEPENDENT VARIABLE, AND THE VALUES ARE PLOTTED AGAINST
THIER INDICES.  IF D IS A MATRIX WITH > 1 ROW, THE FIRST ROW
SHOULD CONTAIN THE VALUES OF THE INDEPENDENT VARIABLE AND
THE REMAINING ROWS SHOULD CONTAIN CORRESPONDING VALUES FOR
THE DEPENDENT VARIABLES.

      THE LEFT ARGUMENT S INDICATES SCALING.  IF OMITTED THE
GRAPH WILL BE SCALED AUTOMATICALLY TO BE 40 TYPEWRITER LINES
HIGH BY 70 TYPEWRITER SPACES WIDE.  IF YOU WISH TO SPECIFY
THE SIZE ONLY AND ALLOW THE PROGRAM TO SCALE AUTOMATICALLY,
THE VALUES OF S SHOULD BE AS FOLLOWS:

   S[1] = THE DESIRED HEIGHT OF THE GRAPH IN LINES (AT LEAST 5)
   S[2] = THE DESIRED WIDTH OF THE GRAPH IN SPACES (AT LEAST 10)

      IF YOU PREFER TO SPECIFY THE SCALING OF THE GRAPH
YOURSELF, YOU MUST GIVE IN ADDITION THE VALUES FOR
THE SCALE ORIGIN AND THE DESIRED DIFFERENCES BETWEEN
SCALE MARKS.  IN THIS CASE S SHOULD BE ASSIGNED
VALUES AS FOLLOWS:

   S[1] = THE DESIRED HEIGHT IN LINES (AT LEAST 5)
   S[2] = THE SCALE ORIGIN FOR THE VERTICAL AXIS
   S[3] = THE DIFFERENCE BETWEEN SCALE MARKS ON THE
          VERTICAL AXIS (SCALE MARKS ARE EVERY 5 LINES)
   S[4] = THE DESIRED WIDTH IN SPACES (AT LEAST 10)
   S[5] = THE SCALE ORIGIN FOR THE HORIZONTAL AXIS
   S[6] = THE DIFFERENCE BETWEEN SCALE MARKS ON THE
          HORIZONTAL AXIS (SCALE MARKS ARE EVERY 10 SPACES)

      IF SEVERAL DEPENDENT VARIABLES ARE PLOTTED, THEY
ARE SHOWN ON THE GRAPH USING SUCCESSIVE SYMBOLS FROM
THE VARIABLE PS.  IF PS DOES NOT EXIST, IT WILL BE CREATED
WITH THE VALUE '○+⎕×⊥⍑⍋∇•'.  IF THE PLOT SYMBOLS IN PS ARE
EXHAUSTED, THE FIRST ARE USED AGAIN.  THE VARIABLE PS MAY
BE REASSIGNED TO CAUSE DIFFERENT PLOT SYMBOLS TO BE USED.
IF TWO OR MORE VARIABLES HAVE A VALUE IN THE SAME PRINT POSITION,
THEIR SYMBOLS WILL BE OVERSTRUCK AT THAT POSITION.
```

IF THE VARIABLES DX AND DY ARE DEFINED TO BE CHARACTER ARRAYS CONTAINING DESCRIPTIONS OF THE INDEPENDENT AND DEPENDENT VARIABLES RESPECTIVELY, THE DESCRIPTIONS WILL BE PRINTED ALONG THE APPROPRIATE AXES. IF THESE ARE EMPTY VECTORS OR DO NOT EXIST NO DESCRIPTION WILL BE PRINTED.

THE RESULT G IS A CHARACTER VECTOR WITH EMBEDDED CARRIAGE RETURNS. IT APPEARS TO BE A MATRIX WITH SHAPE:

```
SIZE+4 13
```

WHERE SIZE IS THE 2-ELEMENT VECTOR SPECIFYING HEIGHT,WIDTH OF THE PLOT IN LINES,SPACES (40 70 IF S NOT SPECIFIED). A FUNCTION 'VTOM' IS PROVIDED TO CONVERT SUCH A VECTOR RESULT TO A MATRIX WITH THE SAME APPEARANCE.

THE FUNCTIONS 'VS' AND 'AND' ARE PROVIDED TO HELP FORM THE RIGHT ARGUMENT TO PLOT. THESE FUNCTIONS ALLOW VECTOR OR MATRIX ARGUMENTS AND FORM A MATRIX RESULT BY JOINING THE ROWS IN THE APPROPRIATE ORDER. FOR EXAMPLE,

```
A AND B AND C VS D
```

WOULD JOIN THE VECTORS A, B, C, AND D TOGETHER TO FORM A MATRIX HAVING ROWS TAKEN FROM D, A, B, AND C, IN THAT ORDER.

THERE IS SUFFICIENT WORKING AREA TO EXECUTE IF

```
⎕WA[1] ≥ (0.4××/SIZE) ⌈ (3××/⍴D)
```

ANOTHER 1100 WORDS CAN BE RELEASED BY USING

```
)ERASE .GRPDOC
```

EXAMPLES:

```
10 20 PLOT ι10
   10+                    o
     |                  o
     |                o
     |              o
     |            o
    5+          o
     |        o
     |      o
     |    o
     |  o
    0+
     /+---------+---------+
      0         5        10
```

```
10 20 PLOT (0,ι10) AND (11-ι11) AND (11ρ5) VS 0,ι10
10++                   o
  |  +               o
  |    +           o
  |      +       o
  |        +   o
 5+□ □ □ □ □ ■ □ □ □ □ □
  |        o   +
  |      o       +
  |    o           +
  |  o               +
 0+o                   +
  /+---------+---------+
   0         5        10
```

```
10 1 5 20 5 5 PLOT ι15
11+            o
  |          o
  |        o
  |      o
  |    o
 6+  o
  |o
  |
  |
  |
 1+
  /+---------+---------+
   5        10        15
```

```
      ∇PLOT [⎕]∇
      ∇SYMBOLS←SIZE PLOT Y;Y;X;IO;A;ERRIF;ORIGIN;INCR;NINCR;MIN;MAX;RANGE;ΔRANGE;PRECISION;INDEX;
INTS;NUMS;FORM;TAG;YTAG;AXIS;STRETCHER;LOCS;VALUES;
[1]    IO←''⍴⎕IO ⍝ USE SCALAR VALUE OF ⎕IO
[2]    A←⍎(,(0=⎕NC'DX DY A PS')≠4 3⍴'DX←DY←0⍴ PS←'),'''○+⎕×⊥⊤Δ∇∘'''⍝ DEFINE DX DY PS
[3]    A←⎕FX 4 11⍴'Y←M ERRIF C   X←C/0   →C↓0   M;'' ERROR'' '⍝ DEFINE ERROR HANDLING FUNCTION
[4]    →'RIGHT ARGUMENT SHAPE'ERRIF 1 2∧.≠⍴⍴Y
[5]    →'RIGHT ARGUMENT EMPTY'ERRIF 0∊⍴Y
[6]    ⍝
[7]    ⍝ DETERMINE INDEPENDENT (X) AND DEPENDENT (Y) VALUES
[8]    ⍝
[9]    Y←(¯2↑1,⍴Y)⍴Y ⍝ MAKE SURE Y IS A MATRIX
[10]   X←(Y[,IO;];⍳¯1↑⍴Y)[IO+1=''⍴⍴Y;]⍝ SELECT X
[11]   Y←(2↑1≠1↑⍴Y)↓Y ⍝ Y IS DEPENDENT VARIABLES
[12]   ⍝
[13]   ⍝ DETERMINE SCALINGS
[14]   ⍝
[15]   ⍎(0=⎕NC'SIZE')/'SIZE←40 70'⍝ DEFAULT PLOT SIZE
[16]   SIZE←,SIZE ⍝ MAKE SURE IT'S A VECTOR
[17]   →'LEFT ARGUMENT LENGTH (MUST BE 2 OR 6)'ERRIF 2 6∧.≠⍴SIZE
[18]   →(6≠⍴SIZE)/L1 ⍝ IF SCALING IS NOT FULLY SPECIFIED
[19]   INCR←SIZE[IO+2 5]
[20]   ORIGIN←SIZE[IO+1 4]
[21]   SIZE←SIZE[IO+0 3]
[22]   ⍝
[23]  L1:→'PLOT SIZE LESS THAN 5 10'ERRIF SIZE∨.<5 10
[24]   NINCR←SIZE÷5 10 ⍝ NUMBER OF INCREMENTS
[25]   →(×⎕NC'INCR')/L2 ⍝ BYPASS AUTOMATIC SCALING
[26]   ⍝
[27]   Y←,Y
[28]   MAX←(⌈/(Y≠⌊/⍳0)/Y),⌈/X ⍝ DISREGARD INFINITIES
[29]   MIN←(⌊/(Y≠⌈/⍳0)/Y),⌊/X ⍝ FOR AUTO SCALING
[30]   RANGE←(MAX-MIN)×SIZE÷SIZE+0.5 ⍝ ALLOW FOR ROUNDING VALUES
[31]   Y←''⍝ TO SAVE SPACE
[32]   →'X OR Y VALUES CONSTANT'ERRIF 0∊RANGE
[33]   INCR←10*0 1⊤10⍟RANGE÷NINCR ⍝ EXPONENT,MANTISSA FOR Y,X
[34]   NUMS←1 1.25 1.5 1.75 2 2.5 3 3.5 4 5 6 7 7.5 8 9 10 12.5 ⍝ MANTISSA VALUES
```

```
[35]  INCR←INCR[IO;]×NUMS[IO+NUMS+.≤1 0↓INCR]⍝ NEAREST NICE INCREMENT
[36]  ΔRANGE←(MIN+INCR×NINCR+0.1 0.05)-MAX ⍝ IN WHICH ORIGIN MAY BE LOCATED
[37]  MIN←MIN+INCR×0.05 0.025 ⍝ LARGEST VALUE THAT ORIGIN COULD HAVE
[38]  PRECISION←10*1+⌊10⍟ΔRANGE⌈1E¯150 ⍝ GRANULARITY OF AXIS LABELING
[39]  PRECISION←PRECISION÷10*ΔRANGE<PRECISION|MIN ⍝ REDUCE IT IF MIN WON'T FIT
[40]  ORIGIN←PRECISION×⌊MIN÷PRECISION ⍝ HAS FEWEST SIGNIFICANT DIGITS
[41]  ⍝
[42]  ⍝ CREATE THE PLOT CHARACTERS
[43]  ⍝
[44] L2:X←⌊0.5+(X-¯1↑ORIGIN)×10÷¯1↑INCR ⍝ X INDEX VALUES
[45]  X←X+1E50×(X<0)∨X>¯1↑SIZE ⍝ MAKE OUT-OF-RANGE LARGE
[46]  Y←⌈(¯0.5+1↑SIZE)-(Y-1↑ORIGIN)×5÷1↑INCR ⍝ UNINVERTED Y INDEX VALUES
[47]  Y←Y+1E50×(Y<0)∨Y>1↑SIZE
[48]  SIZE←SIZE+4 14 ⍝ 3 12 FOR AXES + 1 1 TO RELAX BOUNDS + 0 1 FOR CR'S
[49]  INDEX←⍉(Y×¯1↑SIZE)+(⍴Y)⍴X+12 ⍝ INDICES OF A RAVELLED PLOT MATRIX
[50]  Y←X←''
[51]  SYMBOLS←(¯1↑⍴INDEX)⍴PS
[52]  INDEX←,INDEX×1000
[53]  INDEX←INDEX+(⍴INDEX)⍴⎕AV⍳SYMBOLS ⍝ UNIQUE IDENT OF EACH SYMBOL AT EACH POINT
[54]  INDEX←(INDEX<1E50)/INDEX ⍝ REMOVE OUT-OF-RANGE POINTS
[55]  INDEX←INDEX,(⎕AV⍳'B')+¯1000++\(1↑SIZE)⍴1000×¯1↑SIZE ⍝ PUT CR'S AT ENDS OF LINES
[56]  INDEX←INDEX[⍋INDEX]⍝ REORDER POINTS TO FIND NEIGHBORS
[57]  INDEX←(INDEX≠¯1↓¯1,INDEX)/INDEX ⍝ REMOVE DUPLICATES
[58]  INDEX←INDEX-¯1↓¯1000,INDEX-1000|INDEX ⍝ SYMBOLS AND INTER-POINT DISTANCES
[59]  LOCS←(1000>INDEX)/⍳⍴INDEX ⍝ LOCATIONS OF OVERSTRIKES
[60]  INDEX[LOCS]←1000+INDEX[LOCS]⍝ GIVE THEM UNIQUE LOCATIONS
[61]  STRETCHER←((⍴INDEX)+⍴LOCS)⍴1 1
[62]  LOCS←LOCS+(⍳⍴LOCS)-IO ⍝ NEW LOCS FOR BJ'S (BACKSPACES)
[63]  STRETCHER[LOCS]←0
[64]  INDEX←STRETCHER\INDEX ⍝ MAKE ROOM AND
[65]  INDEX[LOCS]←1000+⎕AV⍳'B'⍝ INSERT BJ'S
[66]  SYMBOLS←⎕AV[1000|INDEX]
[67]  LOCS←('B'=SYMBOLS)/⍳⍴SYMBOLS ⍝ LOCS OF CR'S
[68]  X←(¯13+¯1↑SIZE),⍴,DX ⍝ X PLOT DIMENSION, DESCRIPTION LENGTH
[69]  AXIS←((×¯1↑X)×-12+(1↑X)⌊⌈0.5×+/X)↑(⌊/X)↑,DX ⍝ FINAL X DESCRIPTION
[70]  INDEX←⌊INDEX÷1000 ⍝ INTER-POINT DISTANCES
```

```
[71]   INDEX[(¯3↓LOCS),¯1↑LOCS]←1+(12×¯3↓1=LOCS-¯1↓(IO-1),LOCS),ρAXIS ⍝ NO TRAILING BLANKS
[72]   LOCS←+\INDEX ⍝ ULTIMATE SYMBOL LOCATIONS
[73]   INDEX←''
[74]   STRETCHER←(¯1↑LOCS)ρ0 0
[75]   STRETCHER[LOCS+IO-1]←1
[76]   LOCS←''
[77]   ⍝
[78]   ⍝ DETERMINE AXIS LABELING NUMBERS (TAGS)
[79]   ⍝
[80]   NINCR←⌊NINCR
[81]   VALUES←+\ORIGIN⍪((⌈/NINCR),2)ρINCR ⍝ AXIS VALUES
[82]   VALUES←VALUES×(|VALUES)>(ρVALUES)ρINCR×⎕CT ⍝ SET VERY SMALL VALUES TO 0
[83]   NUMS←((1+1↑NINCR),1)↑VALUES ⍝ Y AXIS VALUES
[84]  L3:FORM←⍕|NUMS ⍝ SMALLEST CHARACTER REPRESENTATION
[85]   INTS←+/∧\'.'∧.≠FORM ⍝ NECESSARY NUMBER OF INTEGER POSISTIONS
[86]   PRECISION←((0⌈(9⌊¯1↑ρFORM)-1+INTS),¯3)[IO+('E'∊FORM)∨INTS>9]⍝ FRACTION POSITIONS
[87]   TAG←(10,PRECISION)⍕NUMS ⍝ FINAL CHAR MATRIX OF TAGS
[88]   →(0≠⎕NC'YTAG')/L4 ⍝ IF YTAG ALREADY PROCESSED
[89]   YTAG←⊖TAG ⍝ ELSE CREATE YTAG
[90]   NUMS←((1+¯1↑NINCR),¯1)↑VALUES ⍝ X AXIS VALUES
[91]   →L3
[92]  L4:SYMBOLS←STRETCHER\SYMBOLS ⍝ FINAL PLOT
[93]   ⍝
[94]   ⍝ INSERT AXIS LABELING
[95]   ⍝
[96]   LOCS←¯1↓IO,1+('B'=SYMBOLS)/⍳ρSYMBOLS ⍝ LOC OF 1ST SYMBOL ON EACH LINE
[97]   SYMBOLS[(¯1↑LOCS)+(⍳ρAXIS)-IO]←AXIS ⍝ INSERT X DESCRIPTION
[98]   AXIS←(¯12↑'/'),(1↑X)ρ'+---------'⍝ X AXIS MARKS
[99]   SYMBOLS[(2↑¯3↓LOCS)∘.+(⍳ρAXIS)-IO]←AXIS,[IO-0.5](ρAXIS)↑'    ',TAG
[100]  LOCS←¯3↓LOCS ⍝ FORGET X AXIS LABELLING AREA
[101]  Y←(¯3+1↑SIZE),ρ,DY ⍝ Y PLOT DIMENSION, DESCRIPTION LENGTH
[102]  AXIS←⌽(1↑Y)ρ'+||⊤⊤'⍝ Y AXIS MARKS
[103]  SYMBOLS[LOCS∘.+0 11]←((0⌊⌊¯0.5×-/Y)⌽(1↑Y)↑,DY),[IO+0.5]AXIS ⍝ Y DESCRIPTION AND MARKS
[104]  SYMBOLS[(('+'=AXIS)/LOCS)∘.+(1-IO)+⍳10]←YTAG
      ∇
```

DISTRIBUTION LIST

	Copy No
Defense Industrial Supply Center	1 thru 3
NADC Technical Library	4 thru 6
NADC (copy 7 for 201) (copy 8 for 202)	7, 8
DTIC	9 thru 20

A4

END

FILMED

1-83

DTIC